Australian Government
Department of Defence
Defence Science and
Technology Organisation

A Simulation Model of Networked Tracking for Anti-Submarine Warfare

J. M. Thredgold, S. J. Lourey, H. X. Vu and M. P. Fewell

Maritime Operations Division
Defence Science and Technology Organisation

DSTO-TR-2372

ABSTRACT

This report describes a simulation model of sonar tracking, developed to explore the effect of networking sonars on tracking performance. The tracker is an extended Kalman filter with data association by nearest-neighbour in Mahalanobis distance. Data fusion algorithms also use Mahalanobis distance. Simulation outputs have been verified against analytical results where possible.

Published by

Maritime Operations Division
DSTO Defence Science and Technology Organisation
PO Box 1500
Edinburgh South Australia 5111 Australia

Telephone: (08) 8259 5555
Fax: (08) 8259 6567

AR-014-686
January 2010

APPROVED FOR PUBLIC RELEASE

A Simulation Model of Networked Tracking for Anti-Submarine Warfare

Executive Summary

This report describes a simulation model of active sonar tracking, developed to explore the effect of networking. The model builds on earlier analytical work (DSTO-TR-2086), which found that detection probabilities as low as 30% could be useful for track initiation if there was a network of sonars such that these detections could be shared with other sonars with similar probabilities of detection.

Having shown analytically that networking offered benefits in terms of starting tracks, the simulation was developed in order to see if this networking advantage carries through to other stages of the tracking process. This report documents the simulation model and the testing of this model against results from the previous analytical work. We find good agreement, thereby providing a level of validation of the simulation model. A study which uses the simulation model to compare centralised and distributed tracking is described in a companion report (DSTO-TR-2373).

Authors

J.M. Thredgold
Maritime Operations Division

Jane Thredgold commenced working in the Maritime Operations Division at DSTO in early 2007, after completing a PhD in Mathematics at the University of South Australia. She is currently performing operations analysis work relevant to anti-submarine warfare.

S.J. Lourey
Maritime Operations Division

Simon Lourey was awarded the BE, MEngSc and PhD degrees in 1990, 1993 and 2005 respectively, by the Electrical Engineering Department of the University of Melbourne. He joined the Maritime Operations Division of the DSTO in 1995 and has worked on various projects in sonar simulation and signal processing. Dr Lourey is a Member of the Institute of Electrical and Electronic Engineers and a Graduate Member of the Institution of Engineers, Australia.

H.X. Vu
Maritime Operations Division

Han Vu joined the Maritime Operations Division at DSTO in 2007. She has completed an Honours degree in Mathematics from the University of South Australia. She is currently working on sonar performance modelling.

M.P. Fewell
Maritime Operations Division

Matthew Fewell joined DSTO in 2001, coming from an academic physics background. He has worked and published in experimental nuclear structure physics, gaseous electronics, atom–photon interactions including coherent effects, laser physics, plasma processing of materials, the theory of network-centric warfare and its modelling at the operational level (including cognitive issues), human-in-the-loop experimentation, and weapon–target allocation in ship air defence. He is at present working on issues in anti-submarine warfare from surface-ship and maritime-patrol perspectives.

Contents

VARIABLES

1. INTRODUCTION 1

2. SCENE GENERATION 2
2.1 Example of Sensor and Target Disposition 2
2.2 Sonar Performance and Operation 3
2.3 Generation of Target Detections 3
2.4 Generation of False (or Non-target) Detections 5
2.5 Sonar Parameters 6

3. TRACKING ALGORITHM 6
3.1 The Kalman Filter 6
3.2 Maintenance State of Tracks 8
3.3 Initiation of Tentative Tracks — the First and Second Measurements 10
3.4 Addition of Third and Subsequent Measurements 11
3.5 Tracking Parameters 11
3.6 Data Fusion 12
3.6.1 Detection Fusion for Centralised Tracking 12
3.6.2 Track Fusion for Distributed Tracking 12

4. COMPARISON OF SIMULATION WITH ANALYTICAL RESULTS 13
4.1 Probability of Track Initiation—Uniform Detection Probability 13
4.1.1 One Sonar 14
4.1.2 Network of Sonars 14
4.2 Probability of Track Initiation — Range-Varying Detection Probability ... 15
4.2.1 Synopsis of Analytical Results 15
4.2.2 Probability of Track Initiation over a Simulation Run 17
4.2.3 Probability of Track Initiation in Five Ensonifications in a Simulation 18

5. SUMMARY AND CONCLUSIONS 20

6. REFERENCES 22

Variables

c	speed of sound in water
$\mathbf{C}_k$	track estimation errors covariance matrix
f	frequency
$\mathbf{F}_k$	state transition matrix
G	Mahalanobis distance
$\mathbf{H}$	observation matrix
$\mathbf{K}_k$	Kalman gain matrix
L	array length
N_{fa}	expected number of false detections per ping
P_d	probability of detection per ensonification
P_{dn}	networked detection probability
P_{dk}	detection probability for sonar k
P_{fa}	probability of a false detection
P_{fti}	probability of false track initiation per 5 consecutive ensonifications
P_{ti}	probability of track initiation per 5 consecutive ensonifications
r	range or distance from a sonar
r_{max}	maximum detection range
$\mathbf{R}_k$	covariance matrix of measurement noise
$\widetilde{q}_T$	manoeuvring index
$\mathbf{Q}_k$	covariance matrix of target motion model
T	time between measurements
$\mathbf{u}_k$	state noise (plant noise) vector represented by random Gaussian process
$\mathbf{v}_k$	measurement noise vector represented by random Gaussian process
v_{max}	maximum speed at which target could be moving
$\mathbf{x}_k$	state vector
$\hat{\mathbf{x}}_{k+1\|k}$	estimate of the state vector at time $k+1$ given measurements up to time k
$\mathbf{z}_k$	measurement vector
β	bandwidth
$\hat{\tau}$	travel time of the sonar pulse
$\hat{\psi}$	bearing
γ	threshold value from chi squared distribution
λ	wavelength
ρ	covariance of two (scalar) random variables
σ	standard deviation of a (scalar) random variable

1. Introduction

This report describes a simulation developed to investigate the effect of networking on tracking in the context of active sonar. The model is an extension of earlier analytical studies [1] which indicated benefits could be obtained through networking, and was developed to allow us to examine the effects of some of the assumptions made in the analytical work. This report gives details of the model and describes its testing against the analytical results. A study of the capability impact of networking sonar systems using the model is described in a companion report [2].

The previous work focussed on track initiation and compared the performance when tracks are formed on detections from a group of sonars (centralised tracking) with the case where each individual sonar forms tracks using only its own detections (distributed tracking). The simple analytical approach taken did not consider measurement errors or the effect of false detections, nor the resulting difficulties in associating detections with each other to form tracks. Effectively, we assumed that, if the target is detected three times in five pings, then we always start a track. In reality this may not always occur, as measurement errors may mean that the detections are not close enough together or a false detection may interfere with the track formation process by confusing the picture. In short, the analytical study doesn't give sufficient consideration to the issue of *data association*.

Also the significant issue of the increased false track rate arising from centralised tracking was ignored. Our previous work included an analytical study of false-detection rate, but it did not address the issue of the false detections leading to false tracks.

The simulation model allows us to investigate the effects of measurement errors and false detections. The false (i.e. non-target) detections in the model are assumed to be 'noise' detections which are not associated with any nearby real object. Recurrent clutter type detections which result from objects, such as bottom features or fish schools, are a separate issue that is not considered here. The simulation model includes data association, track maintenance and track termination, not just the track initiation step. All sonar operation is assumed to be active and multiple monostatic.

The simulation code can be divided into three distinct parts — scene generation, tracking and analysis of the tracks. In the first of these three parts, a scenario is set up and detections are generated. At the end of this stage, there exists a separate file for each sensor in the scenario, containing all of the detections made by that sonar and information about the sonar properties. These detection files are passed to the tracking code, which outputs data structures containing tracks. The third stage of the simulation involves extracting information of interest about the tracks, and providing this in a useful form for further analysis, by writing the data to an Excel spreadsheet.

Section 2 of this report describes the scene generation part of the simulation model, including how detections (both target and non-target) are generated. The next section, Section 3, gives a detailed description of the tracking algorithm, which is based on the Kalman filter. The focus of our work was on analysing the effect of networking, not on developing or improving

tracking algorithms, and so we have used a standard algorithm. The level of detail in this section is provided for those readers who are not familiar with tracking techniques. Section 4 reports some verification of the simulation model by comparing the results obtained with the earlier analytical work [1].

2. Scene Generation

A scenario is set up by editing parameter values in a Matlab script file. The user needs to specify:

- the number of sensors and their location, given as (x,y) coordinate pairs (depth is neglected),
- the target(s) path(s),
- the probability of detection versus range curve for each sensor (the model does not assume all sonars in the network have identical properties, but does ignore bearing dependence and the effect of ship baffles),
- the probability P_{fa} per range-bearing cell per ping of obtaining a false detection (this is assumed to be the same for all sensors, but differences in the size of the range-bearing cells may result in different expected numbers of false alarms per ping for the different sensors; note that there are no Doppler bins—the Doppler dimension is neglected),
- the sonar parameters (bandwidth, frequency and array size) which are used to calculate the expected number of false alarms per ping and in the generation of measurement errors, and
- the ping interval and the duration of the simulation.

With different seeds used for the random number generator, the code produces different detection lists in different runs of the scenario, with the times and number of actual detections (target and false) and the measurement error of those detections all varying. This allows Monte Carlo analysis of a single scenario.

2.1 Example of Sensor and Target Disposition

An example of a scenario is shown in Figure 1. This scenario comprises a network of three sonars. There is a single submarine target present. The positions of the sonars and the submarine at the beginning of the simulation are shown in the figure, together with their relative motion, which is constant throughout the simulation. If the sonars were hull mounted sonars attached to ships, then this scenario could be considered as representative of a task group in the process of running over an adversary submarine. The companion report [2] presents detailed results from running this scenario; the test results presented in Section 4 below use much simpler scenarios.

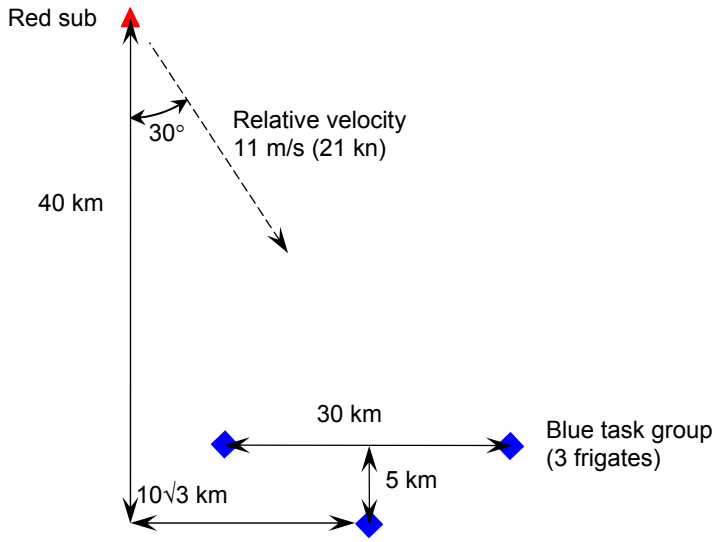

Figure 1: The positions of the three sensors and the submarine at the beginning of the simulation. The relative velocity is constant throughout the simulation.

2.2 Sonar Performance and Operation

Sonar performance in the simulation model is defined by probability of detection versus range curves, with no bearing or depth dependence.(1) The simulation code can be used to model a sonobuoy field or a network of sonar systems attached to vessels, provided appropriate probability of detection versus range curves are chosen. For simplicity and ease of comparison with the analytical study, and to keep the modelling unclassified, in the work reported here we use exponential functions as the probability of detection versus range curves.(2) We also assume multiple monostatic operation of the sonars.

2.3 Generation of Target Detections

Whether or not a sonar ping produces a target detection is determined by a draw from a uniform random distribution compared with the detection probability at the relevant range. If the target is detected, then the recorded position is generated using additional random numbers to introduce measurement error. Measurement errors have a zero-mean bivariate normal distribution in range and bearing.

In our example, each sonar pings once every 60 seconds, which is also the time step of the simulation. The covariance matrix of the errors is set so that the standard deviations correspond to the Cramér–Rao lower bound [3], approximately 1 metre in range and 2.5° in bearing. This requires the component of the Cartesian position error caused by the bearing

(1) Since carrying out the testing described in Section 4, the code has been modified to include both bearing and depth dependence.
(2) It is also possible to use more realistic probability of detection versus range curves obtained from sonar performance modelling, including aspect-dependent target strength.

uncertainty to increase linearly with range, as illustrated in Figure 2. The measurement error covariance matrix affects the size of the data association gates used in the tracker.

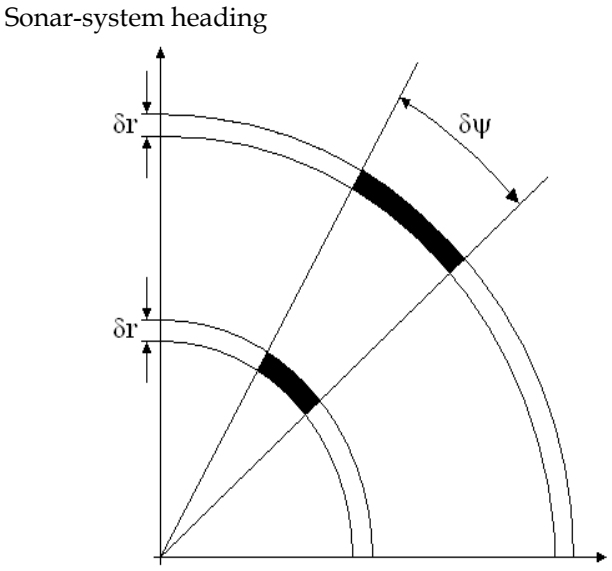

Figure 2: Variation of detection cell size with range

As the aim of the simulation is to evaluate the possible advantages of networking sensors rather than the characteristics of various tracking algorithms, some simplifications are applied in the generation of measurement errors. Strictly speaking, Gaussian random measurement errors should be added to the range and bearing measurements before transformation to Cartesian coordinates, but this means that the measurements will not have a Gaussian distribution in the Cartesian domain. Non-Gaussian measurements would require the use of a more complicated, non-linear tracking algorithm which is not necessary because the critical phenomenon in this case is that measurement errors increase as the range increases. A simple model that represents this while leaving the Kalman filter as a valid tracking solution simulates the measurement by using the actual target position and then adding correlated noise, in the following manner.

The covariance matrix R for a particular measurement at range r and bearing ψ is [4]

$$R = \begin{bmatrix} \dfrac{c^2 \operatorname{var}(\hat{\tau})}{4}\sin^2\psi + \operatorname{var}(\hat{\psi})r^2\cos^2\psi & \left(\dfrac{c^2 \operatorname{var}(\hat{\tau})}{4} - \operatorname{var}(\hat{\psi})r^2\right)\cos\psi\sin\psi \\ \left(\dfrac{c^2 \operatorname{var}(\hat{\tau})}{4} - \operatorname{var}(\hat{\psi})r^2\right)\cos\psi\sin\psi & \dfrac{c^2 \operatorname{var}(\hat{\tau})}{4}\cos^2\psi + \operatorname{var}(\hat{\psi})r^2\sin^2\psi \end{bmatrix} \quad (1)$$

with [3]

$$\operatorname{var}(\hat{\tau}) = \frac{3}{\beta^2\pi^2}, \qquad \operatorname{var}(\hat{\psi}) = \frac{3\lambda^3}{2\pi^2 L^3}, \quad (2)$$

where $\hat{\tau}$ is the estimated travel time for the sonar pulse that resulted in the measurement, $\hat{\psi}$ is the estimated bearing of the detection, β is the bandwidth, λ is the wavelength (c/frequency, c = speed of sound in water) and L is the size (in metres) of the sonar array. The measurement errors in Cartesian coordinates are generated by multiplying the square root of the matrix R by a vector of pseudorandom values. That is, the measurement coordinates are calculated as

$$\begin{bmatrix} x_M \\ y_M \end{bmatrix} = \begin{bmatrix} x_T \\ y_T \end{bmatrix} + R^{1/2} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}$$

where (x_M, y_M) is the recorded measurement, (x_T, y_T) is the actual target position and n_1 and n_2 are values from a pseudorandom Gaussian distribution with mean zero and unit standard deviation.

2.4 Generation of False (or Non-target) Detections

The false detections generated differ for each run of the simulation. The number of false detections occurring at each ping is drawn from a Poisson distribution with the expected value depending on the probability P_{fa} of a false detection per range–bearing cell per ping set by the user, and on the resolution of the sonar as determined from the frequency, bandwidth and size of the array.

The area scanned by the sonar is broken up into cells or segments with the range increment and bearing span specified by the resolution that can be achieved by the sonar. Each of the cells is modelled as having a fixed false alarm probability P_{fa}, but the cells do not have equal area. There should be fewer false alarms per unit area a long way from the sensor than close to the sensor. Figure 2 illustrates the way that two cells which have the same probability of false alarm associated with them correspond to very different areas on the scan.

Generating simulated false alarms requires the expected number of false alarms per scan to be known. Assuming that the beamwidth ($\delta\psi \approx \lambda/L$, the wavelength divided by the array size) is in radians then each scan has $2\pi/\delta\psi$ beams. Also if the sonar is operated to a maximum range $r_{\max}$ then each beam contains $2r_{\max}/c\delta\tau$ cells, where $\delta\tau = 1/\beta$ (β is the bandwidth). The number of false alarms expected from a scan is the number of beams times the number of cells in a beam times the probability that a cell produces a false detection. That is

$$N_{fa} = \frac{4\pi r_{\max}}{c\delta\tau\delta\psi} P_{fa} . \qquad (3)$$

The first step in the generation of false alarms for each ping is to draw a number from a Poisson distribution with this rate to give the number of false alarms for the current scan. These false detections are then distributed in range and bearing over the sonar's field of view by drawing ranges from a uniform distribution on $(0, r_{\max})$ and bearings from the uniform distribution on $(0, 2\pi)$.

The assumption, in this section, that the sonar uses a CFAR detector requires some caution in preparing the probability of detection versus range relationship. It is often assumed in

performance modelling that the threshold the signal must be greater than for detection to be declared is constant. A CFAR detector adjusts the detection threshold to maintain a constant false alarm rate so that modelling assuming a constant detection threshold will be inappropriate unless the background is not varying. This may apply if false alarms are resulting from ambient noise (noise limited conditions) but if the false alarms result from reverberation (reverberation limited conditions) the detection threshold will not be constant because the reverberation level (usually) decreases with range. Correct modelling of a reverberation limited scenario will require a probability of detection versus range curve adjusted for the varying detection threshold.

2.5 Sonar Parameters

By changing the sensor geometry and target path(s) an infinite number of scenarios could be constructed. To investigate how changes in sonar performance affect the outcome of the simulation we can vary the simulation parameters, while keeping the scenario layout the same. These parameters are

- the probability of detection versus range curves for each sonar,
- the probability P_{fa} of false detections per cell per ping for each sonar.

Using different probability of detection curves with the same scenario geometry allows us to effectively model the same scenario under different environmental and acoustic propagation conditions. By varying the probability of false detections we can investigate the effect of high and low clutter environments on the tracking performance.

3. Tracking Algorithm

The tracking algorithm used is relatively simple as our interest is not in improving tracker performance but in comparing the tracks produced when tracking is performed centrally with those obtained by fusing the tracks from individual sonars. The flowchart in Figure 3 provides an overview of the tracking process. The steps within this process will be explained in more detail in coming sections. Readers who are familiar with tracking may not need to read this section in detail — the tracking algorithm we have used is a standard Kalman filter [5] with data association by nearest neighbour in Mahalanobis distance [6]. The 3-in-5 rule, which requires that a track have at least three detections from every five scans to be initiated or retained, is used not only for track initiation, but also for track maintenance and termination.

3.1 The Kalman Filter

Before discussing the steps shown in the flowchart in Figure 3 in more detail, we introduce the Kalman filter [3,7,8]. The Kalman filter is the optimum estimator for the sequence of states produced by the Gauss-Markov model

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{u}_k , \tag{4}$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k , \tag{5}$$

where

$\mathbf{x}_k$ is the state vector of the target at time instant k,
$\mathbf{F}_k$ is the (known) state transition matrix,
$\mathbf{u}_k$ representing uncertainty in state transitions (often called plant noise) is a random Gaussian vector with zero mean and covariance matrix $\mathbf{Q}_k$,
$\mathbf{z}_k$ is a measurement vector,
$\mathbf{v}_k$ representing uncertainty in observations (measurement noise) is a random Gaussian vector with covariance matrix $\mathbf{R}_k$, and
$\mathbf{H}$ is the observation matrix, defined as

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{6}$$

Note that we have dropped the subscript on $\mathbf{H}$ as this matrix does not vary with time. For the tracking application the state vector

$$\mathbf{x}_k = \begin{bmatrix} x_k & \dot{x}_k & y_k & \dot{y}_k \end{bmatrix}^T, \tag{7}$$

contains the position and velocity vectors of the target and a commonly used model of the target dynamics has the transition matrix

$$\mathbf{F} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{8}$$

(again dropping the subscript), which represents a target moving at constant velocity with random perturbations and T is the time interval between measurements. The covariance matrix $\mathbf{Q}$ for the random perturbation is defined as

$$\mathbf{Q} = \tilde{q}_T \begin{bmatrix} T^3/3 & T^2/2 & 0 & 0 \\ T^2/2 & T & 0 & 0 \\ 0 & 0 & T^3/3 & T^2/2 \\ 0 & 0 & T^2/2 & T \end{bmatrix} \tag{9}$$

where $\tilde{q}_T$ is the manoeuvrability index, a parameter which represents the standard deviation of "random" accelerations experienced by the target. (The assumed value of this parameter controls the responsiveness of the Kalman Filter tracker derived from the Gauss Markov model to a manoeuvring target.) The measurement vector $\mathbf{z}_k$ is assumed to be of the form

$$\mathbf{z}_k = \begin{bmatrix} \tilde{x}_k \\ \tilde{y}_k \end{bmatrix} \sim N\left(\begin{bmatrix} x_k \\ y_k \end{bmatrix}, \mathbf{R}_k \right), \tag{10}$$

where the covariance $\mathbf{R}_k$ of the measurement noise is defined as

$$\mathbf{R}_k = \begin{bmatrix} \sigma_x^2(k) & \rho_{xy}(k) \\ \rho_{xy}(k) & \sigma_y^2(k) \end{bmatrix}. \tag{11}$$

The iterative procedure for estimating the new state after each measurement starts from a prediction based on the old state and corrects it based on the measurement. If we do not

obtain a measurement then the predicted state can be used for the estimation of the state vector at the next time.

Assuming we have a track defined up to time k, the prediction of the state at the time k+1 and the error covariance of this prediction is:

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{F}\hat{\mathbf{x}}_{k|k}, \tag{12}$$

$$\mathbf{C}_{k+1|k} = \mathbf{F}\mathbf{C}_{k|k}\mathbf{F}^T + \mathbf{Q}_k. \tag{13}$$

The double subscript notation ($j \mid k$) indicates that this is the estimate of the state or covariance at the time of the jth measurement made from the measurements available at the time of the kth measurement.

If we obtain a measurement that falls within the gate (defined in Section 3.4 below) at the time k+1, then the above estimate is corrected using

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}\left(\mathbf{z}_{k+1} - \mathbf{H}\hat{\mathbf{x}}_{k+1|k}\right), \tag{14}$$

where

$$\mathbf{K}_{k+1} = \mathbf{C}_{k+1|k}\mathbf{H}^T\left(\mathbf{H}\mathbf{C}_{k+1|k}\mathbf{H}^T + \mathbf{R}_k\right)^{-1} \tag{15}$$

is a matrix called the Kalman gain. The estimates are Gaussian random vectors so complete characterisation requires computation of the error covariance

$$\mathbf{C}_{k+1|k+1} = \mathbf{C}_{k+1|k} - \mathbf{K}_{k+1}\mathbf{H}\mathbf{C}_{k+1|k}. \tag{16}$$

A track consists of an estimate of the target state $\hat{\mathbf{x}}_k$ and the covariance matrix $\mathbf{C}_{k|k}$ which describes the error in that estimate.

3.2 Maintenance State of Tracks

Within the model, there are four possibilities for the maintenance state of a track. A track can be either old or current, and either tentative or confirmed; the four maintenance states consist of all possible combinations of these two pairs. An old track is one which has not been updated by at least three measurements within the last five detection opportunities. A tentative track contains only one or two measurements; a confirmed track has three or more.

The maintenance state of a track may change from tentative to confirmed, and from current to old, but confirmed tracks can not become tentative, nor old tracks current again. Once a track is marked as old, it is no longer checked for association with new measurements and its maintenance state will not change. That is, an old tentative track can not become a confirmed track.

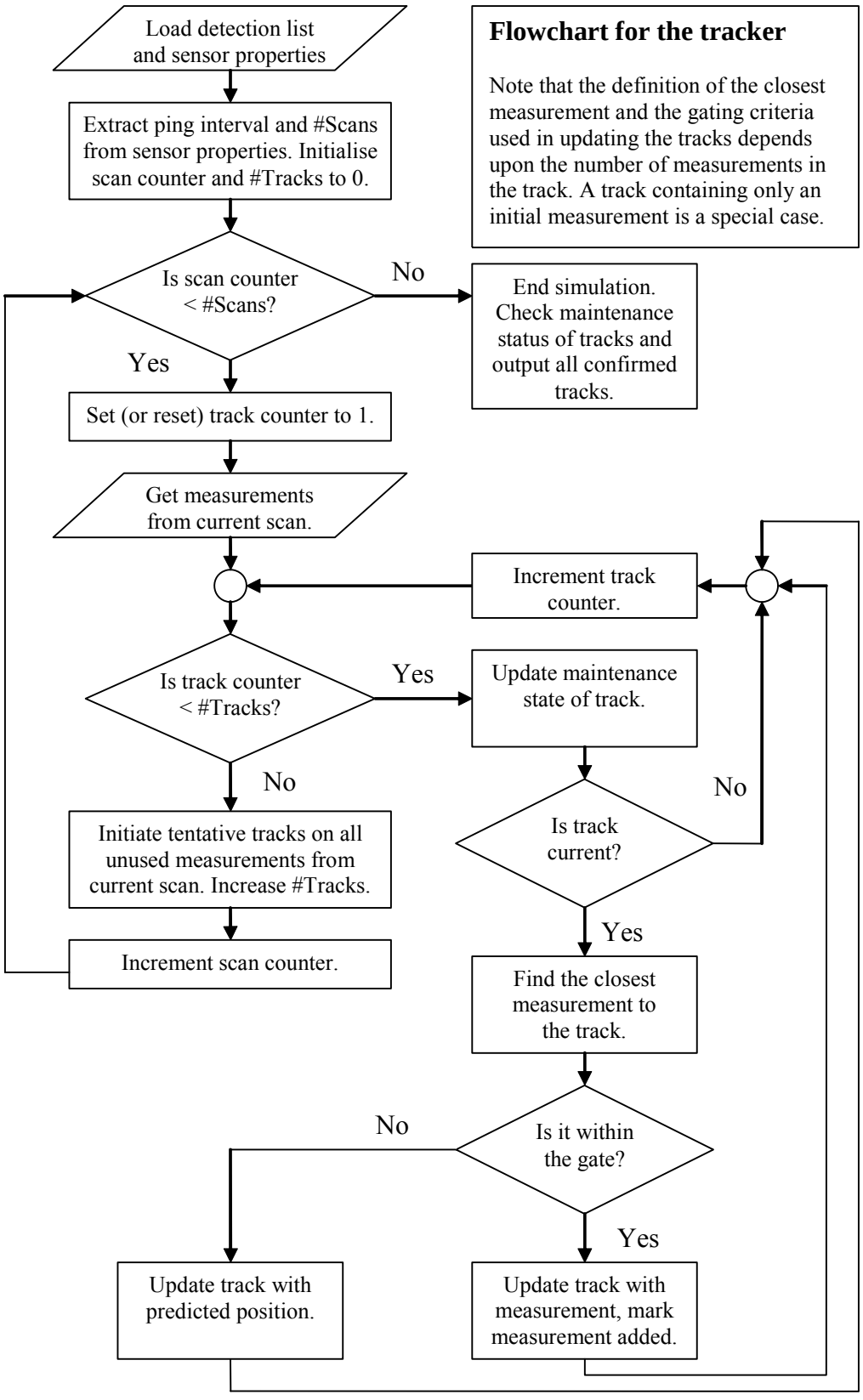

Figure 3: Flowchart for tracking algorithm

3.3 Initiation of Tentative Tracks — the First and Second Measurements

Any detection that is not associated with an existing track is considered as a potential source of a new track. Given an isolated detection, the algorithm will search for a second detection within the next three time steps using an expanding, almost square, gate centred on the initial detection. The size of the gate is primarily determined by the length of time elapsed since the initial detection and the maximum relative velocity with which it is estimated potential targets could be travelling [5, p. 213]. With T representing the time since the initial measurement and $v_{\max}$ the maximum feasible target speed, the gate is centred on the initial point and has sides of length $2v_{\max}T$, plus a small allowance for the errors in both the initial measurement and the measurement we are prospectively adding to the track. These errors usually differ in the x and y directions (which is why we describe the gate as *almost* square) and are calculated by taking the square root of the diagonal elements of the respective covariance matrices. As each measurement will have a different error covariance matrix the gate is slightly different for each measurement.

If R_1 is the covariance matrix of the first measurement (x_1, y_1) taken at time T_1, and R_2 is the covariance matrix of the measurement (x_2, y_2) taken at T_2 which we are considering for addition to the track, then the gate, centred at (x_1, y_1) will have sides of length

$$2\left(v_{\max}\left(T_2 - T_1\right) + \sigma_x(1) + \sigma_x(2)\right) \tag{17}$$

in the x direction, and

$$2\left(v_{\max}\left(T_2 - T_1\right) + \sigma_y(1) + \sigma_y(2)\right) \tag{18}$$

in the y direction.

If more than one measurement falls within the box gate the closest measurement, in terms of Cartesian distance, is used. If there is no measurement within the gate obtained at a detection opportunity the track is not updated with a prediction, as with only a single measurement we cannot predict in which direction, or at what speed, the target is moving. The gate will expand for the next detection opportunity.

Once we have two detections in a track, we can estimate the position and velocity of the possible target. When a second measurement is added to a track, the state vector for the track can be calculated. The state vector for the above example would be

$$\hat{\mathbf{x}}_{2|2} = \begin{bmatrix} \hat{x}_2 \\ \hat{\dot{x}}_2 \\ \hat{y}_2 \\ \hat{\dot{y}}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \left(x_2 - x_1\right)/T \\ y_2 \\ \left(y_2 - y_1\right)/T \end{bmatrix}, \tag{19}$$

with the covariance matrix for the estimate of the error in the track calculated as:

$$\mathbf{C}_{k+i|k+i} = \begin{bmatrix} \sigma_x^2(2) & \frac{\sigma_x^2(2)}{T} & \rho_{xy}(2) & \frac{\rho_{xy}(2)}{T} \\ \frac{\sigma_x^2(2)}{T} & \frac{\sigma_x^2(2)+\sigma_x^2(1)}{T^2} & \frac{\rho_{xy}(2)}{T} & \frac{\rho_{xy}(2)+\rho_{xy}(1)}{T^2} \\ \rho_{xy}(2) & \frac{\rho_{xy}(2)}{T} & \sigma_y^2(2) & \frac{\sigma_y^2(2)}{T} \\ \frac{\rho_{xy}(2)}{T} & \frac{\rho_{xy}(2)+\rho_{xy}(1)}{T^2} & \frac{\sigma_y^2(2)}{T} & \frac{\sigma_y^2(2)+\sigma_y^2(1)}{T^2} \end{bmatrix}, \tag{20}$$

where $T = T_2 - T_1$.

3.4 Addition of Third and Subsequent Measurements

Once a track contains two measurements, the state vector and covariance matrix which form that track will be updated at every detection opportunity, either with a measurement (using equations (12) through to (16)), or using the predicted values, (equations (12) and (13) only), until the track fails the 3-in-5 rule and is marked as old.

When there are more than two measurements in a track, we update the track with the measurement that has minimum Mahalanobis distance [6, p. 36],

$$\mathrm{G}(\mathbf{z}_k) = \left(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}\right)^T \left(\mathbf{H}\mathbf{C}_{k|k-1}\mathbf{H}^T + \mathbf{R}_k\right)^{-1} \left(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}\right), \tag{21}$$

provided that this measurement falls within an elliptical gate centred on the predicted position of the track. We test whether a measurement falls within the gate by checking whether the Mahalanobis distance of the measurement from the track satisfies the condition

$$\mathrm{G}(\mathbf{z}_k) < \gamma \tag{22}$$

where γ is a threshold determined from the chi squared distribution with two degrees of freedom. We used 95% as the probability value — with the resulting threshold value of 5.991. The size of the gate is dependent upon the assumed manoeuvrability of the target as defined by the parameter $\tilde{q}_T$ of the Kalman filter, the current uncertainty of the track as measured by the covariance matrix $\mathbf{C}_{k|k}$, and the error covariance matrix $\mathbf{R}_k$ of the detection being checked for association with the track. The gate may therefore differ in size for each measurement considered.

Our algorithm allows a detection to associate with any track to which it is the closest; that is, a detection may be associated with more than one track, but any track will gain no more than one detection per ping.

3.5 Tracking Parameters

As for the simulation of target states and measurements, there are a number of parameters which affect the performance of the tracker. These parameters are

- $\tilde{q}_T$, which controls the responsiveness of the tracker to a manoeuvring target,
- $v_{\max}$, the assumed maximum relative velocity of the target, and

- the track initiation and track maintenance rules.

For optimal performance of the tracker, the assumed maximum velocity and target manoeuvrability should be varied according to the behaviour of the target(s) present in the scenario. These values, along with the measurement error covariance matrices, affect the size of the gates in which points must fall to initiate and maintain a track. If the values of these parameters are too low, the target will not be tracked. If the values are too high, the target tracking should not be adversely affected, but a higher number of false tracks may result.

3.6 Data Fusion

The earlier analytical study [1] examined two tracking options: centralised and distributed. To produce comparable results with the simulation, the same detections — both target and false —from each simulation run are used by the two tracking options. The only difference is in how these detections are processed to form tracks, as described below.

3.6.1 Detection Fusion for Centralised Tracking

For centralised tracking, the detection lists from each sensor are combined and passed to the tracker as a single set. The combination rule is the equivalent of a simple logical 'or', except for the case where two detections occur at the same time very close to one another. Detections from two sonars are fused into a single detection if the Mahalanobis distance between them is consistent with zero at the 95% confidence level, as determined by a χ^2 test with 2 degrees of freedom. That is, two measurements $z_1(x_1,y_1)$ and $z_2(x_2,y_2)$ with error covariance matrices $\mathbf{R}_1$ and $\mathbf{R}_2$ respectively will be fused if

$$\left(\mathbf{z}_2 - \mathbf{z}_1\right)^T \left(\mathbf{R}_1 + \mathbf{R}_2\right)^{-1} \left(\mathbf{z}_2 - \mathbf{z}_1\right) < \gamma \,, \tag{23}$$

where γ is the same threshold value determined from the chi squared distribution with two degrees of freedom used when deciding whether a measurement falls within the elliptical gate described in Section 3.4. If the measurements are fused, the combined measurement is calculated as

$$\mathbf{z} = \mathbf{R}_2\left(\mathbf{R}_1 + \mathbf{R}_2\right)^{-1}\mathbf{z}_1 + \mathbf{R}_1\left(\mathbf{R}_1 + \mathbf{R}_2\right)^{-1}\mathbf{z}_2 \,, \tag{24}$$

with the combined covariance matrix

$$\mathbf{R} = \mathbf{R}_1\left(\mathbf{R}_1 + \mathbf{R}_2\right)^{-1}\mathbf{R}_2 \,. \tag{25}$$

This approach to measurement fusion was chosen to avoid an extensive rewriting of the single sensor tracker to incorporate multi-sensor operation. In the future, the conceptually more elegant approach of passing all measurements to the tracker will be applied.

3.6.2 Track Fusion for Distributed Tracking

For distributed tracking, the tracking algorithm is applied separately to the detections recorded by each sensor. The tracks are then fused into a single list. If two sensors are tracking the target at the same time, and the track position estimates are close enough together, the tracks will be fused into a single track. For example if sonar 1 has a track on the target from ping 10 to ping 14 and sonar 2 has a track on the target from ping 12 to ping 17, for analysis

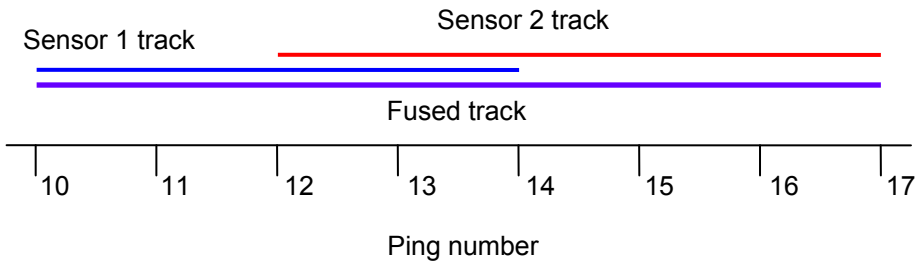

Figure 4: Illustration of track fusion for distributed tracking

purposes this is regarded as a single track on the target from ping 10 to ping 17, as Figure 4 illustrates. As with detection fusion in the centralised tracking case, a χ^2 test is used to determine whether two tracks should be fused. We follow the method of Bar-Shalom & Li [5, pp. 440–3] assuming tracks are independent.

4. Comparison of Simulation with Analytical Results

A number of tests were performed to check that the results from the simulation are consistent with the previous analytical results [1] when the assumptions matched.

4.1 Probability of Track Initiation—Uniform Detection Probability

To check that the simulation gives the same values for the probability of track initiation as the theoretical work we ran a simplified version of the simulation for just five pings. The probability of detection for all sensors was defined as uniform over the sensor's field of view, so that the position of the target does not affect the results. We also set the measurement errors to zero and switched off the generation of false alarms. By performing multiple runs (each of only five pings in length) and recording whether or not a target track was formed in each run, we were able to calculate the probability of starting a track as the percentage of runs in which a track was initiated. This allows us to compare the results from the simulation with the theoretical curve calculated from the equation [1]:

$$P_{\mathrm{ti}} = \sum_{j=p}^{q} \binom{q}{j} P_{\mathrm{d}}{}^{j} \left(1 - P_{\mathrm{d}}\right)^{q-j} \tag{26}$$

where j enumerates the number of detections in the q consecutive ensonifications. The above equation assumes statistical independence of successive ensonifications, but so does the simulation model. Evaluating this formula for the 3-in-5 rule we used in the simulation gives

$$P_{\mathrm{ti}} = P_{\mathrm{d}}{}^{3}\left(10 - 15P_{\mathrm{d}} + 6P_{\mathrm{d}}{}^{2}\right). \tag{27}$$

Note that both the theory and the simulation model implement the 3-in-5 rule as requiring 3 detections in 3 distinct ping cycles. So in a network of 3 sensors a track will not be initiated if all sensors detect the target in one ping cycle but there are no other detections.

4.1.1 One Sonar

The simplest case considers just one sonar. Figure 5 shows that the results from the simulation without measurement errors and false detections are in good agreement with the theoretical results for the idealised case where data association is not a problem — there are neither measurement errors nor non-target detections to confuse the picture. Repeating the analysis with measurement error included (blue triangles in Fig. 5) we can see that this results in a slight decrease in the probability of track initiation, as we would expect. However adding false detections with $P_{fa} = 10^{-5}$ (which corresponds to an expected value of 7.8 false detections per ping), seems to have little effect on the probability of target track initiation — the green crosses in Figure 5 essentially coincide with the red squares.

4.1.2 Network of Sonars

In our earlier analytical work we also calculated theoretical curves showing the probability of track initiation for a network as opposed to a single sensor, to illustrate the possible benefits of networking. If m independent sonars emit one ensonification each, then the probability that at least one of them makes a detection—the 'networked detection probability' P_{dn}—is

$$P_{dn} = 1 - \prod_{k=1}^{m} (1 - P_{dk}), \tag{28}$$

where k enumerates the sonars and P_{dk} is the detection probability for sonar k.[3] This provides another type of detection probability for use in Equation (27), and allows us to

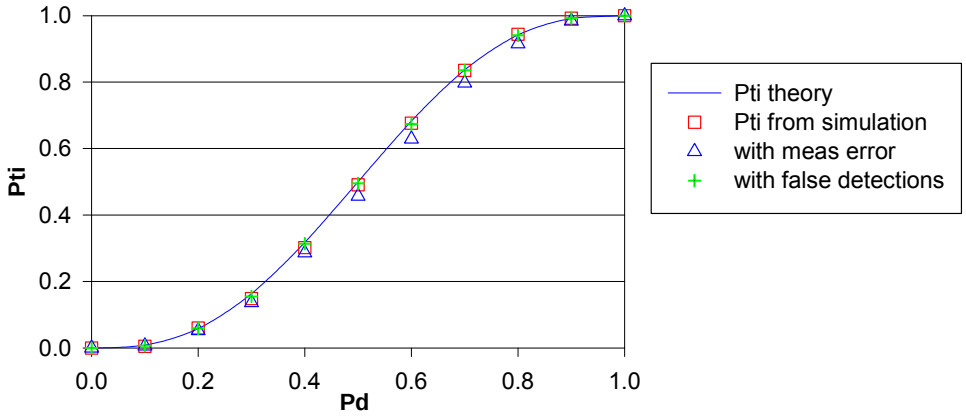

Figure 5: Probability of target track initiation versus probability of detection: theoretical curve and results for simulation without measurement errors and false detections, with measurement error (but no false detections) and with false detections (but no measurement errors).

[3] Equation (28) assumes multiple monostatic operation, where each sonar processes returns only from its own ensonifications, and that the sonars do not interfere with each other. Multistatic processing might give even greater increase in P_{dn} and hence P_{ti}.

produce curves of the probability of track initiation by a network of *m* sensors. Running the simplified simulation model with multiple sensors also produces results in good agreement with the theoretical curves, as shown in Figure 6, where the curves for the simulation results are virtually indistinguishable from the theoretical ones.

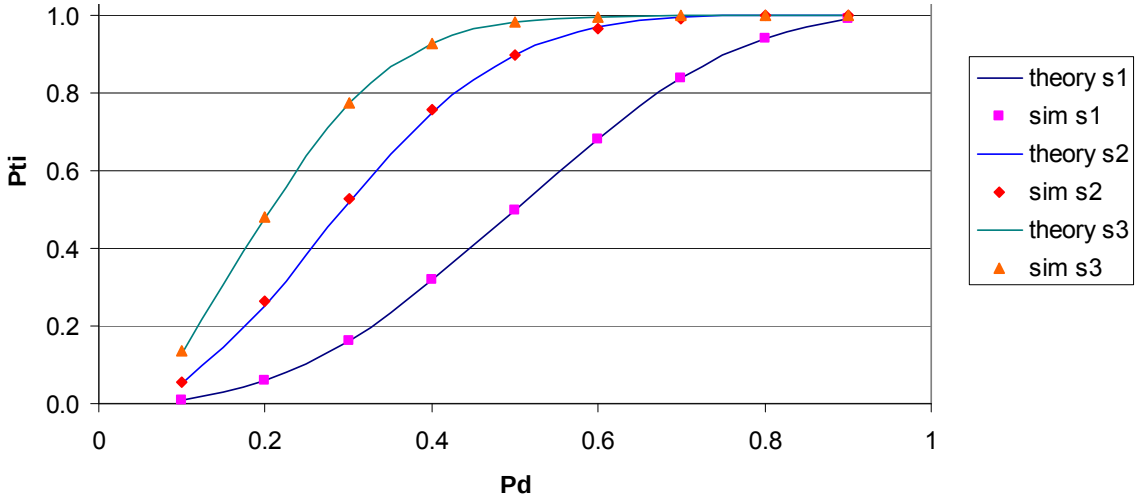

Figure 6: Probability of networked (centralised) track initiation versus probability of detection, theoretical curves (solid lines) and results for simulation without measurement error (points), for networks of 2 and 3 sensors. The single sensor curve, identical to that shown in Figure 5, is also shown for comparison.

4.2 Probability of Track Initiation — Range-Varying Detection Probability

To go beyond this simple analysis where the networked probability of track initiation was calculated as a function of the probability of detection, it is necessary to allow the detection probability to vary with range.

4.2.1 Synopsis of Analytical Results

In the earlier analytical work [1] two types of functions were used to model the range dependence of detection probability. These were an exponential form,

$$P_{\mathrm{d}}(r) = P_0 \exp\left(-r/a\right), \tag{29}$$

to represent the case of a long, low probability tail, and a Fermi-function,

$$P_{\mathrm{d}}(r) = P_0 \frac{1+\exp\left(-b/a\right)}{1+\exp\dfrac{r-b}{a}}, \tag{30}$$

to model the effects of a sharp cut-off, while avoiding the extremes of a definite-range law ('cookie-cutter').

The analytical study found little benefit in networking when the second option was used for the probability of detection curves. For a P_d with a definite-range law (i.e. a 'cookie-cutter' shape with $P_d = 1.0$ inside the detection range), there can be no networking gain from sharing detections. This is because a sonar with this shape of P_d has

- 100% detection probability within its detection range, and so needs no assistance with detecting things in this region, and
- zero detection probability outside its detection range, and so cannot provide assistance with detections there.

It is the low-probability tail of the exponential shape that creates the opportunity for networking advantage, through the combination of low-probability regions from several sensors. This argument relies on the value of P_d inside the detection range being 100%. When a lesser value applies, then the opportunity for networking advantage from sharing detections again arises [1]. We believe this conclusion to be robust against choice of metric.

Due to the result from the analytic study, we do not use Fermi-functions in the simulation. Where simple representative probability of detection curves are used, we have chosen the exponential form.

Figure 7 shows a typical plot produced in the analytic study. There are three sonars, positioned on an equilateral triangle, and each was assumed to have an exponential probability of detection curve with $P_0 = 1.0$ and $a = 1.5$ km. The contours show the areas in which we would expect to start a track using the 3-in-5 rule, with 80 and 95% probability. The dashed lines show the distributed tracking case, in which each sonar individually performs tracking using just its own detections. This corresponds to data fusion at the track level. That is, sonars do not report until a track is formed, and tracks are then fused. If, in contrast, we assume data fusion at the detection level, with centralised track initiation (i.e. using Eq. 28 in Eq. 27), then the outer, labelled, contours are obtained.

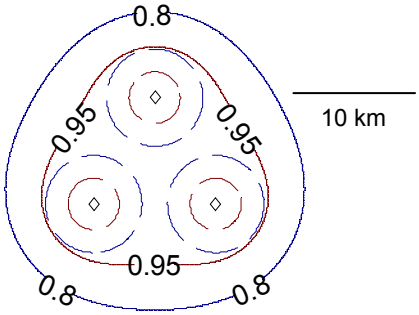

Figure 7: The dashed lines show the contours of 80% (blue) and 95% (red) track-initiation probability P_{ti} per group of 5 ensonifications for 3 sonars spaced 10 km apart, each performing track initiation separately using only its own detections. The exponential detection probability curve with $P_0 = 1$ and $a = 1.5$ km is assumed. The solid, labelled lines are the contours of networked track-initiation probability where detections from all the ensonifications are shared and tracking is performed centrally.

4.2.2 Probability of Track Initiation over a Simulation Run

As a further check on the simulation code, an attempt was made to reproduce this contour plot from simulation results. Unlike the analytic work where we calculate the theoretical probability that we would detect a target *if* one was present, the simulation requires a target to actually *be* present. The simulation was run for a number of target paths, starting on a circle with radius R centred at the midpoint of the sonars and coming in at different angles towards that centre point (i.e. along the radii of the circle). Figure 8 shows some possible target paths.

For this scenario, the three sonars are each assumed to have the same exponential probability of detection curve with $P_0 = 1.0$ and $a = 1.5$ km. We run the simulation without false detections or measurement errors, in order to keep the simulation as close as possible to the assumptions under which the analytical results were obtained.

Using the rotational symmetry of the sonar layout to reduce the number of cases, we ran the scenario 100 times with each of the different target paths, which were spaced at angle increments of $\pi/12$. The time at which the first target track was initiated was recorded for each of these runs and this figure converted to a distance. The 80 and 95% 'contours' for centralised tracking for the simulation results were calculated by taking the distance by which 80 or 95% of the runs had started a track. The results are shown in Figure 9 for R values of 27 km and 20.5 km. They clearly depend on the starting position of the target (i.e. the value of R).

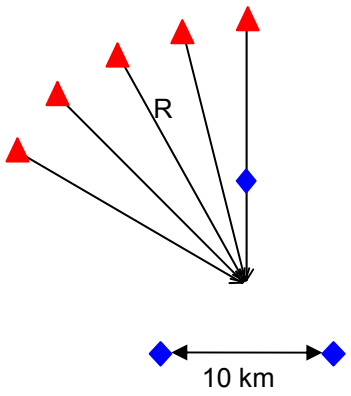

Figure 8: Position of the three sensors (blue diamonds) and the enemy submarine (one of the red triangles) at the beginning of a simulation run. As the simulation progresses, the submarine tracks radially inward at v = 11.25 m/s. This is equivalent to approximately 21 knots. A conventional submarine is unlikely to maintain such a high speed for an extended time, however we regard this value as a relative velocity, representing the fact that the three sensors (assumed to be attached to ships) are also moving.

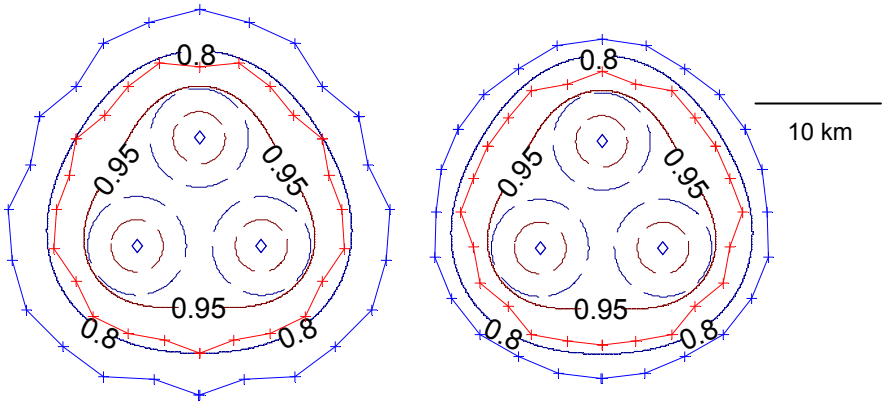

Figure 9: As in Figure 7 but with the simulation centralised tracking 'contour' plots overlaid, as brighter blue (80%) and red (95%) lines with crosses at the data points. The left plot is for R = 27 km and the right plot for R = 20.5 km. The target was moving at the same speed, and the sonars pinging at the same rate, for both plots.

The simulation contours are outside of the analytic ones due to the cumulative effect of the 'lead-in' time that the simulation has. The analytic contours show the area within which a target (if present) would be detected with 80 or 95% probability in exactly five pings. The simulation contours show the area within which the entire simulation (not just five pings) will start a track on a target, which is following a radial path towards the centre, with 80 or 95% probability. This cumulative effect is something that must be kept in mind when viewing the results of simulation runs.

4.2.3 Probability of Track Initiation in Five Ensonifications in a Simulation

In the previous section we attempted to compare whole simulation runs with analytic results. However, the outputs from the simulation are very dependent on the chosen target path and on the length of time for which the simulation runs. Running a simulation over a longer time frame means that cumulative effects are seen in the results. To avoid these difficulties we ran another series of test simulation runs for just five pings, with a single stationary target positioned at a variety of points.

For each target location, the simulation was run one hundred times, each run stopping after just five pings. The number of these runs in which a track is initiated on the target gives us an approximate value for the probability of track initiation at this location. Following the radial fan target path approach taken in the previous section, target locations were specified on a polar coordinate grid centred at the midpoint of the sonars, which were again positioned on the points of an equilateral triangle.

The results from these simulation runs are presented in a different way to the previous results. Instead of drawing a contour plot with lines linking the points at which the probability of track initiation is 80 and 95%, a coloured dot is plotted at each target location, as shown in Figure 10. (Note that some of the points in the figure are obtained by utilising the rotational symmetry of the sensor layout). The colour of the dots signifies the track initiation probability,

with all areas in which the track initiation probability is 80% or higher marked by a green dot. The darker green indicates that the probability is 95% or higher.

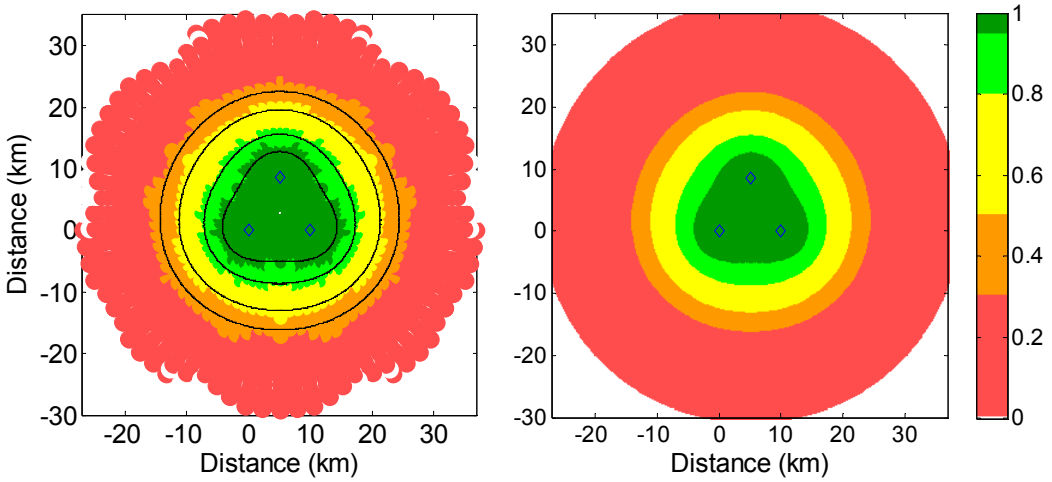

Figure 10: Track initiation probability (centralised tracking) for five pings with a stationary target. The left plot shows the simulation results, the right plot is the analytical (exact) equivalent. The black lines on the left plot are the edges of the analytic contours.

This dot plot approach approximates a filled contour plot for the simulation results. The theoretical filled contour, with the same sonar layout and colour scheme, is also shown in Figure 10 for comparison.

Considering that the simulation results are based on only one hundred runs for each point, the simulation results show reasonable agreement with the analytic work. This is confirmed by the difference plot, shown in Figure 11. The maximum absolute value of the difference between the simulation and analytic probabilities of track initiation is 0.1014, and the mean value is approximately 0.0125. The scale on the colour bar of the contour plot, along with the positive value of the mean, indicates that the simulation appears to start tracks with slightly higher probabilities than the analytic work suggests should be the case, but we believe this to be simply a result of the randomness within the simulation model.

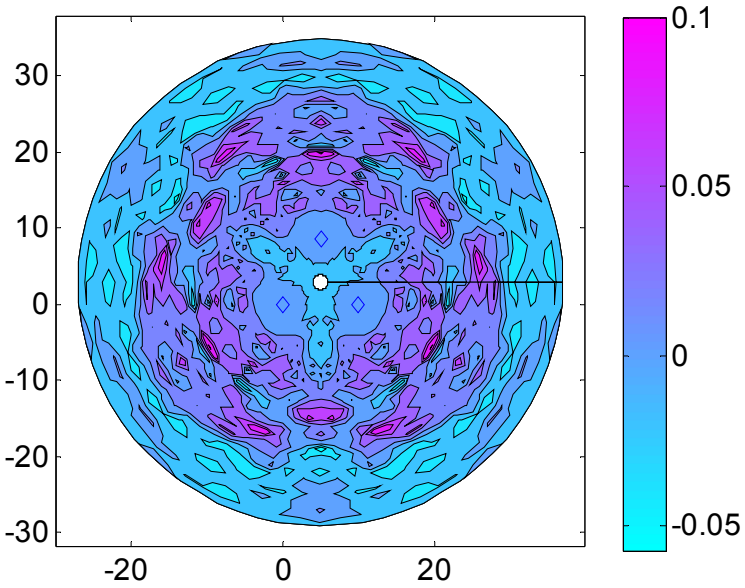

Figure 11: Contour plot of the simulation results minus the analytic values calculated at the same points

5. Summary and Conclusions

This report documents a simulation model of networked tracking in anti-submarine warfare. The model was developed as a tool for exploring the impact of networking on tracking performance in sonar systems. This follows previous analytical work [1] indicating that passing detections to a centralised tracker significantly increases the area over which a target will be tracked compared to the situation in which each sonar in the network performs tracking separately and then pools tracks. The earlier study suggested that significantly improved performance can be obtained with a small network; in fact networking just two sonars gives a substantial improvement. However, this previous work focussed on track initiation, without considering the rest of the tracking process. It also did not analyse the effect of false detections beyond a consideration of false-detection rate, nor the difficulties in associating new detections with existing tracks. The purpose of the present work is to develop a simulation model that can address all of these issues. This report describes the model; the results of using it to explore the capability impact of networking in ASW are described elsewhere [2].

Because the purpose is to explore networking, rather than tracker development, it was not necessary to use a sophisticated tracking algorithm. The sonars generate true detections of any submarines present, subject to specified detection probability versus range behaviour, and false detections that are randomly distributed over their fields of view. The tracker is a standard Kalman filter with data association by nearest neighbour in Mahalanobis distance. All detections, true and false, are treated equally by the tracker. The '3 detections in 5 consecutive pings' rule is used both for track initiation and track termination. Potentially

crossing tracks are handled by allowing a detection to be associated with more than one existing track. When pooling detections from several sonars, we tested for and eliminated duplicate detections using Mahalanobis distance. In the distributed-tracking case, we fused tracks from the sonars to give a single track list.

Three scenarios were used to check the operation of the simulation against the analytical results. First, we considered a single sonar with constant detection probability (i.e. P_d independent of range). When false detections and measurement errors are both switched off, the simulation gives the same values of track-initiation probability P_{ti} as the analytical formula. Switching on false detections does not have a discernable effect on P_{ti}, but the inclusion of measurement error lowers P_{ti} slightly. We attribute this to the occasional failure to recognise three detections as associated with each other, owing to the scatter produced by the measurement errors.

The second check on the operation of the simulation again used constant P_d, but with two or three sonars networked together and tracking centrally. Once again, the simulation values match the analytical formulas for networked P_{ti}.

For the third check, we adopted the exponential behaviour of P_d with range and computed contours of P_{ti} produced by a distributed network of 3 sonars. This test is less clear-cut because of statistical fluctuations, but the comparison with the analytical results is nevertheless satisfactory.

We conclude that the simulation model reproduces the analytical results where the scenario is constructed to match the assumptions underlying the analytical work. This therefore provides a sound basis for extending the analytical results to performance metrics that go beyond track initiation, and including effects of false detections and measurement uncertainty. Results of a first such study are presented in a companion report [2].

6. References

1 Fewell, M.P., Thredgold, J.M. and Kershaw, D.J. (2008) 'Benefits of Sharing Detections for Networked Track Initiation in Anti-Submarine Warfare', Technical Report DSTO-TR-2086 of the Defence Science and Technology Organisation.

2 Thredgold, J.M and Fewell M.P. (2010) 'Distributed versus Centralised Tracking in Networked Anti-Submarine Warfare', Technical Report DSTO-TR-2373 of the Defence Science and Technology Organisation.

3 Burdic, W.S. (1991) *Underwater Acoustic System Analysis*, 2nd Edition, Prentice Hall, Englewood Cliffs NJ.

4 Lourey, S.J. (2004) *Topics in Underwater Detection,* PhD Thesis, The University of Melbourne, Melbourne.

5 Bar-Shalom, Y. and Li, X.-R. (1995) *Multitarget–Multisensor Tracking: Principles and Techniques*, YBS, Storrs CT.

6 Duda, R.O., Hart, P.E. and Stork, D.G. (2001) *Pattern Classification* (2nd edn), Wiley, New York.

7 Bar-Shalom, Y. and Fortman, T. (1988) *Tracking and Data Association*, Academic Press, Orlando FD.

8 Poor, H. V. (1994) *An Introduction to Signal Detection and Estimation*, 2nd Edition, Springer, New York.

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
DOCUMENT CONTROL DATA

1. PRIVACY MARKING / CAVEAT (OF DOCUMENT)

2. TITLE

A Simulation Model of Networked Tracking for Anti-Submarine Warfare

3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)

Document	(U)
Title	(U)
Abstract	(U)

4. AUTHOR(S)

J. M. Thredgold, S. J. Lourey, H. X. Vu and M. P. Fewell

5. CORPORATE AUTHOR

DSTO Defence Science and Technology Organisation
PO Box 1500
Edinburgh South Australia 5111 Australia

6a. DSTO NUMBER	6b. AR NUMBER	6c. TYPE OF REPORT	7. DOCUMENT DATE
DSTO-TR-2372	AR-014-686	Technical Report	January 2010

8. FILE NUMBER	9. TASK NUMBER	10. TASK SPONSOR	11. NO. OF PAGES	12. NO. OF REFERENCES
2008 / 1082714	ERP 07 / 310	CMOD	22	8

13. URL on the World Wide Web

http://www.dsto.defence.gov.au/corporate/reports/DSTO-TR-2372.pdf

14. RELEASE AUTHORITY

Chief, Maritime Operations Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CITATION IN OTHER DOCUMENTS Yes

18. DSTO RESEARCH LIBRARY THESAURUS http://web-vic.dsto.defence.gov.au/workareas/library/resources/dsto_thesaurus.shtml

Antisubmarine warfare, Networked active sonar, Tracking, False alarms

19. ABSTRACT

This report describes a simulation model of sonar tracking, developed to explore the effect of networking sonars on tracking performance. The tracker is an extended Kalman filter with data association by nearest-neighbour in Mahalanobis distance. Data fusion algorithms also use Mahalanobis distance. Simulation outputs have been verified against analytical results where possible.